Finsler Active Contours

John Melonakos, *Student Member*, *IEEE*, Eric Pichon, *Member*, *IEEE*,
Sigurd Angenent, *Member*, *IEEE*, and Allen Tannenbaum, *Member*, *IEEE*

Abstract—In this paper, we propose an image segmentation technique based on augmenting the conformal (or geodesic) active contour framework with directional information. In the isotropic case, the euclidean metric is locally multiplied by a scalar conformal factor based on image information such that the weighted length of curves lying on points of interest (typically edges) is small. The conformal factor that is chosen depends only upon position and is in this sense isotropic. Although directional information has been studied previously for other segmentation frameworks, here, we show that if one desires to add directionality in the conformal active contour framework, then one gets a well-defined minimization problem in the case that the factor defines a Finsler metric. Optimal curves may be obtained using the calculus of variations or dynamic programming-based schemes. Finally, we demonstrate the technique by extracting roads from aerial imagery, blood vessels from medical angiograms, and neural tracts from diffusion-weighted magnetic resonance imagery.

Index Terms—Directional segmentation, Finsler metric, dynamic programming, active contours, diffusion weighted imagery.

1 Introduction

Geodesic active contours [1], [2] have proven to be a very useful tool for a number of segmentation tasks. Basically, the idea is to define an active contour model based on the theory of conformal metrics and on euclidean curve-shortening evolution. This type of curve evolution defines the gradient direction to be that for which a curve will shrink as fast as possible relative to its euclidean arc length. One multiplies the euclidean arc length by a conformal factor defined by the features of interest that one wants to extract and then one computes the corresponding gradient evolution equations. The features that one wants to capture therefore lie at the bottom of a potential well to which the initial contour will flow. The key point is that the conformal structure defines a Riemannian metric in the plane for which the features of interest appear as closed geodesic curves.

In this paper, motivated by certain problems in pattern detection and medical imaging, we develop a version of geodesic active contours in a *Finsler metric* [3], [4]. See our discussion in Section 3 below for the formal mathematical definition. The basic idea is that we add directionality now to the active contours that allows for the segmentation of image data in oriented domains. As alluded to above, isotropic active contour models have been used to segment image data in isotropic domains, meaning that the value of each voxel depends only upon its position in the domain and *not* upon an associated direction. However, in oriented domains, image data depends *both* upon position and direction. In other words, for each position and direction in the domain, there exists a unique voxel intensity. It is here that the concept of the Finsler metric becomes crucial. In fact, if one desires to add directionality to the geodesic active contour framework, we show that the Finsler condition is necessary to ensure that the flow is well posed. Note that the Riemannian metric satisfies the Finsler conditions and is well posed. We show that there are some applications for which the Finsler metric outperforms the Riemannian metric, but certainly, there are others for which the Riemannian metric may be more desirable. For oriented domains, as long as the metric satisfies the Finsler condition, the choice of a particular metric is subject to the given application.

Flows relative to anisotropic metrics have been studied in the mathematics and physics literature; see [5], [6] and the references therein. A very simple directional flow was proposed in some of our earlier work; see [7].

There are many applications of image data in oriented domains. Examples include diffusion-weighted magnetic resonance imaging in which the magnetic field is biased in several directions in order to measure the water diffusivity of biological tissue. In this case, for each position in the tissue and for each direction of the bias field, the corresponding image intensity provides a measure of water diffusivity at that position and direction.

Furthermore, this technique may be used in pattern detection. Consider a small image pattern patch, which we desire to match to our image. Through translations and rotations of the patch throughout the image, we can evaluate a measure of similarity between the patch and the image for the given patch position and direction. Thus, this is also a problem in an oriented domain because for each position and rotation of the patch, there exists a unique measure of similarity.

Geodesic active contours in the Finsler framework provide a mechanism for the minimization of energy functionals defined on oriented domains. We derive both the curve evolution and dynamic programming based implementations for Finsler active contours. The latter is necessary since we will want to consider the evolution of open curves for which the level set methodology is not

- *J. Melonakos and E. Pichon are with Department of Electrical and Computer Engineering, Georgia Institute of Technology, 777 Atlantic Drive, Atlanta, GA 30332-0250. E-mail: jmelonak@ece.gatech.edu, epichon@gmail.com.*
- *S. Angenent is with the Mathematics Department, University of Wisconsin, Madison, Madison, WI 53706. E-mail: angenent@math.wisc.edu.*
- *A. Tannenbaum is with the Departments of Electrical and Computer and Biomedical Engineering, Georgia Institute of Technology, 777 Atlantic Drive, Atlanta, GA 30332-0250. E-mail: tannenba@ece.gatech.edu.*

Manuscript received 12 Aug. 2006; revised 5 Feb. 2007; accepted 25 Apr. 2007; published online 6 June 2007.
Recommended for acceptance by G. Sapiro.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0598-0806.
Digital Object Identifier no. 10.1109/TPAMI.2007.70713.

appropriate. Preliminary results for directional-dependent segmentations may be found in [8], [9].

The idea of using Finsler type metrics for various purposes is of course not original in this work. First of all, regarding curve shortening, Gage [6] has considered curvature driven flows in a Minkowski space. General mean curvature flows relative to Finsler metrics are studied in [10]. Gurtin and Angenent have proposed the use of anisotropic Finsler flows for problems in crystal growth in [5]. In the computer vision literature, such directionally dependent metrics have appeared in [11], [12], [13], [14], [15]. In some nice work, the connections of graph cuts and such metrics have been described in [13], [14], [15]. Geodesic active contours and graph cut methods have been combined in [13], [14]. Further, in [14], the explicit connection between Finsler distances and the flux methods in [12] is considered in some detail.

This paper continues the above line of research. Here, we describe Finsler flows in a completely continuous setting valid both for both open and closed curves embedded in a euclidean space of *any* dimension. The key observation is that if one defines a conformal active contour flow for a direction-dependent conformal factor, then in order for the flow to be well defined, one needs the standard Finsler convexity condition (see Section 3). Without this condition, the flow will be a backward heat equation. Thus, this present work extends the results in [1], [2] who consider conformally euclidean metrics that only depend on position and are in this sense "isotropic." (In this paper, we will follow the standard terminology of the mean curvature flow literature in which "isotropic" flows are defined relative to a Riemannian metric, whereas "anisotropic" flows are defined relative to a direction-dependent Finsler metric; see [3].) It is important to note that while one can get "directionality" in the Riemannian framework for image segmentation by a suitable choice of metric (ellipses have directionality), nevertheless, we believe that the Finsler geodesic active contour approach gives a natural way of performing segmentation in oriented domains.

We now summarize the contents of this paper. In Section 2, we review the theory of energy minimizing flows and geodesic active contours, as well as dynamic programming. Section 3 is the key part of this paper. Here, we define the notion of a "Finsler metric" and derive the geodesic active contour flow relative to such a structure. In Section 4, we describe the dynamic programming based solution and the numerical implementation of such an approach. In Section 5, we show the results of the experiments using these techniques on both MRI tractography and pattern detection applications. In Section 6, we draw some conclusions and describe some future research directions. Finally, we have included two mathematical appendices. The first justifies the use of dynamic programming in our situation in which we have a data driven anisotropic conformal factor, and the second gives another derivation of the Finsler geodesic flow, which also captures some of its interesting properties.

2 Energy Minimizing Curves

Energy minimization approaches to image segmentation have been very popular; see [16], [17], [18], [19], and the references therein. These approaches allow one to define a meaningful energy for a given application and to systematically construct contours that minimize the energy. In this section, we describe two of the key approaches for the minimization of such energy functionals: conformal active contours (based on gradient descent) and dynamic programming. The former works for closed curves, whereas the latter method is valid for curves in which we fix seed and target regions as well.

2.1 Geodesic (Conformal) Active Contours

In the conformal (or geodesic) active contour model, a local cost, $\psi : \mathbf{R}^2 \to \mathbf{R}^+$, is defined based on image information [1], [2]. For a given curve Γ, the total cost $\mathcal{L}(\Gamma)$ is defined as the integration of local costs along the curve

$$\mathcal{L}_\psi(\Gamma) \triangleq \int_\Gamma \psi(\Gamma) \mathrm{d}s. \tag{1}$$

This energy can be interpreted as the ψ-weighted length of the curve. Minimal curves will therefore tend to go through regions where ψ is small while, at the same time, constraining the total conformal euclidean length to be as small as possible. Convergence of this flow is studied in [1], [2]. It is important to note that s is the arc-length parameterization and, therefore, this energy is purely geometric.

If the curve is closed or has fixed end points, a partial differential equation is obtained by calculus of variations that continuously deforms an initial curve $\Gamma(t = 0)$ in a way that optimally minimizes its total cost $\mathcal{L}$. This can be interpreted as a gradient descent on the infinite dimensional space of curves.

In the case of the functional (1), the PDE that deforms a given curve in order to minimize the energy as fast as possible in the L_2 sense is

$$\frac{\partial \Gamma}{\partial t} = -(\nabla \psi \cdot \mathbf{N})\mathbf{N} + \psi \Gamma_{ss}, \tag{2}$$

where $\mathbf{N}$ denotes the unit inward normal.

As is the standard, this may be implemented using level set methods [17], [18].

2.2 Dynamic Programming

Mortensen et al. [20] have proposed the *live-wire segmentation* technique that also determines optimal curves for the same kind of functional. Their framework is based on dynamic programming and is applicable to curves with one end fixed in a given seed region S.

The underlying principle of dynamic programming is the *principle of optimality* verified by minimum-cost problems such as (1) (assuming an optimal curve exists). The principle is that any subpath p of an optimal path P is itself optimal (otherwise, the P could be improved by following another subpath p' instead of p). This leads to the definition of the *value function* $\mathcal{L}^*$, which is the minimal cost to reach the seed region S from any point x of the domain

$$\mathcal{L}^*(x) \triangleq \min\{\mathcal{L}(\Gamma), \Gamma(0) = x, \Gamma(1) \in S\}.$$

In problems such as (1), the value function satisfies the Eikonal equation $|\nabla \mathcal{L}^*(x)| = \psi(x)$ with boundary condition $\mathcal{L}^* = 0$ on S. This equation can be solved numerically using the fast marching algorithm [18], [21] or can be discretely approximated using Dijkstra's algorithm.

From any point in the domain, an optimal curve in the sense of (1) can then be determined by gradient descent on the scalar field $\mathcal{L}^*$.

3 Geodesic Active Contours in a Finsler Metric

In this section, we introduce the notion of direction-dependent active contours. This will be seen to be essentially a version of active contours defined relative to a Finsler rather than a Riemannian metric. If one thinks of a Riemannian metric as being defined by a continuously varying family of inner products on the tangent bundle of a given manifold, a Finsler metric is given by a continuously varying family of Banach space norms. The strict convexity property given below is then an expression of the fact that these norms must satisfy the triangle inequality. More mathematical details about Finsler flows may be also found in Appendix B.

For an excellent exposition of the Finsler property and comparisons to the Riemannian structure, we refer the interested reader to that in [4]. Finally, we should note that versions of curve shortening relative to the Finsler structure have been studied in [5], [6], [22], [23].

3.1 Evolving Space Curves

In this section, we set up our notation and define the notion of *Finsler metric*.

Therefore, consider families of evolving curves of the form $\Gamma : [0,1] \times [0,T) \to \mathbf{R}^n$. For any curve $\Gamma(x,t)$, we denote

$$\mathbf{T} = \frac{\Gamma_x}{|\Gamma_x|}, \quad \frac{\partial}{\partial s} = \frac{1}{|\Gamma_x|}\frac{\partial}{\partial x}, \quad ds = |\Gamma_x| \mathrm{d}x.$$

The curvature vector of Γ is

$$\mathbf{K} = \Gamma_{ss} = \frac{\partial^2 \Gamma}{\partial s^2}.$$

We say that the curve *evolves normally* if

$$\mathbf{V} = \frac{\partial \Gamma}{\partial t} \perp \mathbf{T}$$

holds always. For such curve evolutions, one has

$$\begin{aligned} &\partial_t \mathbf{T} = \partial_s \mathbf{V}, \quad \left[\frac{\partial}{\partial t}, \frac{\partial}{\partial s}\right] = (\mathbf{K} \cdot \mathbf{V}) \frac{\partial}{\partial s}, \text{ and} \\ &\frac{\partial}{\partial t} ds = -(\mathbf{K} \cdot \mathbf{V}) \mathrm{d}s. \end{aligned} \tag{3}$$

For any given function

$$\psi : \mathbf{R}^n \times S^{n-1} \to \mathbf{R}^+,$$

we let

$$\mathcal{L}(\Gamma) = \int_0^L \psi(\Gamma, \mathbf{T}) ds = \int_{x=0}^1 \psi\left(\Gamma, \frac{\Gamma_x}{|\Gamma_x|}\right) |\Gamma_x| \, \mathrm{d}x,$$

where L is the length of Γ. The infinitesimal length function ψ is only defined on unit vectors, but one can extend it naturally to all vectors by requiring it to be positively homogeneous of degree 1. We denote this extension by

$$F(p,v) = |v| \psi\left(x, \frac{v}{|v|}\right)$$

so that the anisotropic length of Γ is

$$\mathcal{L}(\Gamma(\cdot,t)) = \int_0^1 F(\Gamma, \Gamma_x) \, \mathrm{d}x. \tag{4}$$

Because of the homogeneity of F, that is,

$$F(p, tv) = tF(p,v) \text{ for all } x, v \in \mathbf{R}^n \text{ and } t \geq 0$$

the anisotropic length is invariant under orientation preserving reparametrizations of the curve [4]. However, $\mathcal{L}(\Gamma)$ may change if one reverses the orientation of Γ.

The extended anisotropic length function $F(p,v)$ is a never a strictly convex function of v, because it is homogeneous of degree 1. If $F(p,v)^2$ is strictly convex, then F defines a *Finsler metric* on $\mathbf{R}^n$ [3], [4]. A necessary and sufficient condition for this to occur is that $\nabla^2_\theta F(p,\theta)$ should be positive definite on the subspace $\{v \in \mathbf{R}^n : v \perp \theta\}$. We compute this second derivative at the particular vector $\theta = (1, 0, \ldots, 0)$ in terms of ψ. (See [24] for a proof.)

Lemma 1. *If $f : S^{n-1} \to \mathbf{R}$ is a C^2 function and if $F(v) = |v| f(v/|v|)$, then for any $v \in S^{n-1}$ and any pair of tangent vectors $X, Y \in T_v S^{n-1}$, one has*

$$\nabla^2_{X,Y} F(v) = (X,Y) f(v) + \nabla^2_{X,Y} f(v),$$

where $\nabla^2 f$ is the second covariant derivative of $f : S^{n-1} \to \mathbf{R}$.

If $v \neq 0$ is not necessarily a unit vector, then one has

$$\nabla^2_{X,Y} F(v) = \frac{1}{|v|} \left\{ (X,Y) f(v) + \nabla^2_{X,Y} f(v) \right\}.$$

It follows that ψ defines a Finsler metric if and only if the quadratic form defined by $g_{ij}\psi + \nabla_i \nabla_j \psi$ is positive definite.

3.2 First Variation

We can now compute the first variation of our anisotropic length functional and derive the flow for the Finsler geodesic active contours. In this discussion, we assume that the curves are closed or that the family of curves under consideration has fixed end points. Assuming the curve Γ evolves normally, one has

$$\begin{aligned} \frac{d}{dt}\mathcal{L}(\Gamma) &= \frac{d}{dt} \int \psi(\Gamma, \mathbf{T}) \mathrm{d}s \\ &= \int \left\{ \mathbf{V} \cdot \psi_p(\Gamma, \mathbf{T}) + (\partial_t \mathbf{T}) \cdot \psi_v(\Gamma, \mathbf{T}) \right. \\ &\quad \left. - \psi(\Gamma, \mathbf{T}) \mathbf{K} \cdot \mathbf{V} \right\} \mathrm{d}s, \end{aligned}$$

where ψ_p and ψ_v denote derivatives with respect to the first and second variables in $\psi(p,v)$. The derivative with respect to $v \in S^{n-1}$ is a covariant derivative. We use (3) to conclude

$$\begin{aligned} \frac{d}{dt}\mathcal{L}(\Gamma) &= \int \left\{ \mathbf{V} \cdot \psi_p(\Gamma, \mathbf{T}) + \mathbf{V}_s \cdot \psi_v(\Gamma, \mathbf{T}) \right. \\ &\quad \left. - \psi(\Gamma, \mathbf{T}) \mathbf{K} \cdot \mathbf{V} \right\} ds \\ &= \int \left\{ \mathbf{V} \cdot \psi_v(\Gamma, \mathbf{T}) - \mathbf{V} \cdot \partial_s(\psi_v(\Gamma, \mathbf{T})) \right. \\ &\quad \left. - \psi(\Gamma, \mathbf{T}) \mathbf{K} \cdot \mathbf{V} \right\} \mathrm{d}s \\ &= -\int \mathbf{V} \cdot \left\{ \partial_s\big(\psi_v(\Gamma, \mathbf{T})\big) + \psi(\Gamma, \mathbf{T}) \mathbf{K} - \psi_p(\Gamma, \mathbf{T}) \right\} \mathrm{d}s \end{aligned}$$

so that you get the steepest descent with

$$\mathbf{V} = \left[\partial_s(\psi_v(\Gamma, \Gamma_s)) - \psi_p(\Gamma, \Gamma_s)\right]^\perp + \psi(\Gamma, \Gamma_s) \Gamma_{ss}. \tag{5}$$

Here, $X^\perp$ denotes the component of X that is perpendicular to $\Gamma_s = \mathbf{T}$.

Note that $\psi_v(\Gamma, v) \in T_v S^{n-1}$ is a vector perpendicular to v since it is the gradient of a function on S^{n-1} at the point $v \in S^{n-1}$. If you expand the derivative $\partial_s(\psi_v(\Gamma, \Gamma_s))$ you will get two terms, one of which contains second derivatives of Γ, namely, $\psi_{vv}(\Gamma, \Gamma_s) \cdot \Gamma_{ss}$.

The steepest descent flow, then leads to the following quasi-linear PDE

$$\begin{aligned}\mathbf{V} = &\big[\Gamma_s \cdot \nabla_{vp}\psi(\Gamma, \Gamma_s) - \psi_p(\Gamma, \Gamma_s)\big]^{\perp} \\ &+ \big[\psi(\Gamma, \Gamma_s) + \psi_{vv}(\Gamma, \Gamma_s)\big]\Gamma_{ss}.\end{aligned} \tag{6}$$

Here, $\psi_{vv}(\Gamma, \mathbf{T})$ is the linear map on $T_v S^{n-1}$ defined by the second covariant derivative of $\psi(\Gamma, v)$. Thus, for any pair of vectors X, $Y \in T_v S^{n-1}$, one has by definition

$$\nabla^2_{X,Y}\psi(p, v) = \big(X, \psi_{vv} \cdot Y\big).$$

One sees that (6) is a parabolic equation exactly when ψ defines a Finsler metric. This equation defines our model for the *Finsler geodesic active contours*.

The above derivation works for closed curves. In the planar case, one may implement such a flow using level set techniques. We, however, are also in interested direction-dependent flows for curves in which we fix seed and target regions, and for this, we will propose (in Section 4) the use of dynamic programming. This is essential for diffusion tensor imaging in which we want to discover white matter tracts starting from some point in the image.

Finally, in Appendix B below, we derive the first variation of the Finsler functional in terms of the homogeneous extension F, which leads to another numerical scheme.

4 Direction-Dependent Dynamic Programming

In this section, we show how dynamic programming can be used to determine optimal curves. The Finsler metric condition on the anisotropic factor ψ will be assumed throughout this discussion (so that optimal paths will indeed exist).

4.1 Optimal Control and the Principal of Optimality

Consider the optimal control problem of determining a trajectory $x : [0, 1] \to \mathbf{R}^n$ that is optimal with respect to the functional

$$J(x(\cdot), \boldsymbol{u}(\cdot)) = \int_0^1 L(x(t), \boldsymbol{u}(t))\mathrm{d}t.$$

We assume in the discussion below that L is homogeneous of degree 1 in the $\boldsymbol{u}$ variable. The control $\boldsymbol{u}(\cdot)$ is defined by

$$\dot{x}(t) = \boldsymbol{u}(t).$$

For any given starting point x_0, define the value function as the minimum cost for reaching a seed region $S \subset \mathbf{R}^n$ from x_0

$$J^*(x_0) = \inf_{\boldsymbol{u}(\cdot), x(0)=x_0, x(1)\in S} J(x(\cdot), \boldsymbol{u}(\cdot)).$$

When an optimum exists, it may be found using Bellman's *principle of optimality* [25]. Basically, this states that if $x^*(\cdot)$ is an optimal trajectory, then all subpaths are also optimal. This can be expressed by the following relation:

$$J^*(x_0) = \inf_{\boldsymbol{u}(\cdot), x(0)=x_0, x(1)\in S} \left\{ \int_0^r L(x(t), \boldsymbol{u}(t))\mathrm{d}t + J^*(x(r)) \right\}.$$

This means that if an optimal trajectory $x^*(\cdot)$ is found such that $x^*(0) = x_0$ and $x^*(1) \in S$, then for any $r \in]0, 1[$, the subtrajectories $x^*_{|[0,r]}$ and $x^*_{|[r,1]}$ are also optimal. See [25] for a detailed proof.

In our case, using $x_t = \boldsymbol{u}$ and $x(0) = x_0$, the following Hamilton-Jacobi-Bellman equation is obtained:

$$0 = \inf_{\boldsymbol{u}(0)} \{L(x_0, \boldsymbol{u}(0)) + \nabla J^*(x_0) \cdot \boldsymbol{u}(0)\}. \tag{7}$$

In general, the value function may not be differentiable. In that case, the differential (7) holds in the sense of *viscosity theory*. See [26].

Under our above assumptions, this can be applied to the Finsler cost functional

$$\begin{aligned}\mathcal{L}(\Gamma) &= \int_0^L \psi(\Gamma(s), \Gamma_s(s))\mathrm{d}s \\ &= \int_0^1 F(\Gamma, \Gamma_x/|\Gamma_x|)|\Gamma_x|\mathrm{d}x = \int_0^1 F(\Gamma, \Gamma_x)\mathrm{d}x,\end{aligned} \tag{8}$$

where s is arclength and L is length of the curve.

The resulting Hamilton-Jacobi-Bellman equation is

$$0 = \inf_{\Gamma_x(0)} \left\{ \psi\big(\Gamma(0), \frac{\Gamma_x(0)}{|\Gamma_x(0)|}\big)|\Gamma_x(0)| + \nabla\mathcal{L}^*(\Gamma(0)) \cdot \Gamma_x(0) \right\}$$

and, finally,

$$\begin{cases} 0 = \inf_{\hat{\boldsymbol{d}}\in S^{n-1}} \{\psi(\boldsymbol{p}, \hat{\boldsymbol{d}}) + \nabla\mathcal{L}^*(\boldsymbol{p}) \cdot \hat{\boldsymbol{d}}\}, \\ \mathcal{L}^*(s) = 0 \text{ for } s \in S, \end{cases} \tag{9}$$

where anticipating our discussion in Section 5 for images, we denote the given voxel location (that is, point in $\mathbf{R}^n$) as $\boldsymbol{p}$ and direction as $\hat{\boldsymbol{d}}$.

4.2 Numerics

This equation can be solved numerically in a straightforward manner. Several numeric approaches may be used, such as those given in [27], [28], [29]. We use the Fast Sweeping approach proposed in [29]. From any point $\boldsymbol{p}_0 \in \mathbf{R}^n$, an optimal path in the sense of (8) can then be determined by following locally the vector $\hat{\boldsymbol{d}}^*$ for which the minimum is attained in (9).

Algorithm 1. Sweeping algorithm to solve the Hamilton-Jacobi-Bellman (9); see [28]

Require: seed region S, direction-dependent local cost ψ

1: Initialize $\mathcal{L}^*(\cdot) \leftarrow +\infty$, except at starting points $s \in S$, where $\mathcal{L}^*(s) \leftarrow 0$
2: **repeat**
3: **sweep** through all voxels $\boldsymbol{p}$, in all possible grid directions
4: $\hat{\boldsymbol{d}}' \leftarrow \arg\min_{d\in\mathbf{S}^{n-1}} f_{\mathcal{L}^*,\psi}(\boldsymbol{p}, \hat{\boldsymbol{d}})$
5: **if** $f_{\mathcal{L}^*,\psi}(\boldsymbol{p}, \hat{\boldsymbol{d}}') < \mathcal{L}^*(\boldsymbol{p})$ **then** $\mathcal{L}^*(\boldsymbol{p}) \leftarrow f_{\mathcal{L}^*,\psi}(\boldsymbol{p}, \hat{\boldsymbol{d}}')$ and $\hat{\boldsymbol{d}}^*(\boldsymbol{p}) \leftarrow \hat{\boldsymbol{d}}'$ **end if**
6: **end sweep**
7: **until** convergence of $\mathcal{L}^*$

The algorithm sweeps through all points $\boldsymbol{p}$ in search of the least expensive direction. The cumulated cost to reach $\boldsymbol{p}$ from direction $\hat{\boldsymbol{d}}$ is

$$f_{\mathcal{L}^*,\psi}(\boldsymbol{p}, \hat{\boldsymbol{d}}) \triangleq \left(\sum_{k=0}^{n-1} \alpha_k \mathcal{L}^*(\boldsymbol{p} + \boldsymbol{\delta}_k) + \psi(\boldsymbol{p}, \hat{\boldsymbol{d}})\right) \Big/ \left(\sum_{k=0}^{n-1} \alpha_k\right),$$

where the n neighbors $\boldsymbol{p}+\boldsymbol{\delta}_0,\ldots,\boldsymbol{p}+\boldsymbol{\delta}_{n-1}$ of $\boldsymbol{p}$ in direction $\hat{\boldsymbol{d}}$ are interpolated using the components of the vector $\boldsymbol{\alpha}\triangleq[\boldsymbol{\delta}_0|\ldots|\boldsymbol{\delta}_{n-1}]^{-1}\hat{\boldsymbol{d}}$. Thus, in three dimensions, this would be $n=3$ neighbors among 26. If we take then, for example, $\hat{\boldsymbol{d}}=(0.912,0.228,0.342)^t$, one could choose the three neighbors $\boldsymbol{\delta}_0=(1,0,0)^t$, $\boldsymbol{\delta}_1=(1,0,1)^t$, $\boldsymbol{\delta}_2=(1,1,1)^t$, and the corresponding weights would be $\boldsymbol{\alpha}=(\alpha_0,\alpha_1,\alpha_2)=(0.228,0.114,0.570)$. One interpretation is that the value for reaching $\boldsymbol{p}$ from direction $\hat{\boldsymbol{d}}$ will be influenced most by the value at $\boldsymbol{p}+\boldsymbol{\delta}_2$, which is the neighbor as much in direction $\hat{\boldsymbol{d}}$ as the grid allows. Since, in general, it is not exactly in that direction, the final result will also be interpolated using the two other most aligned neighbors $\boldsymbol{p}+\boldsymbol{\delta}_0$ and $\boldsymbol{p}+\boldsymbol{\delta}_1$.

One sees that if the continuous direction $\hat{\boldsymbol{d}}$ is exactly defined by one of the neighboring voxels, that is, $\hat{\boldsymbol{d}}=\boldsymbol{\delta}_k/\|\boldsymbol{\delta}_k\|$, then $f=\mathcal{L}^*(\boldsymbol{p}+\boldsymbol{\delta}_k)+\psi(\boldsymbol{p},\hat{\boldsymbol{d}})\|\boldsymbol{\delta}_k\|$, which is the cost for reaching voxel $\boldsymbol{p}$ from voxel $\boldsymbol{p}+\boldsymbol{\delta}_k$. This same quantity would be computed in Dijkstra's algorithm. Unlike Dijkstra's algorithm, however, the search for the optimal direction is not restricted to discrete grid directions, and the minimization is performed continuously over the sphere S^{n-1}. In our implementation, the minimization is performed over 100 directions sampled uniformly on the sphere,[1] and the coefficients $\boldsymbol{\alpha}(\hat{\boldsymbol{d}})$ are precomputed. Mathematical details and a convergence proof are available in [28].

Note that the number of iterations (where one iteration is defined as one set of all possible directional sweeps) required for convergence depends upon the number of turns in the curve. Each iteration, therefore, can flow information through one complete turn of the curve. In the Applications and Simulations in Section 5, we found all the curves in three iterations or less.

5 Applications and Simulations

Having developed the theory behind Finsler active contours, we now illustrate these via several experiments. First, we demonstrate curve shortening with respect to certain anisotropic conformal factors as opposed to isotropic curve shortening. Second, we provide a synthetic example that demonstrates a particular case where Finsler active contours capture a corner in directional data. Third, these methods are applied to a pattern detection problem, specifically to detect roads and vessels in 2D imagery. Fourth, we show 3D results of these techniques applied to diffusion-weighted magnetic resonance imagery for white matter brain tractography.

5.1 Closed Curves Evolving According to the Finsler Flow

In this section, in order to compare the proposed direction-dependent framework to the isotropic framework, we examine the evolution of a closed bean-shaped curve with respect to three separate conformal factors.

In order to isolate the effect of directional information, we study local costs that do not depend on position but only on the direction $\mathbf{N}=[n_1\ n_2]^t$ (we use the unit normal instead of the unit tangent in defining the conformal factors; for planar curves, this is clearly equivalent):

1. $\psi=1$.
2. $\psi=\max(\frac{1}{\sqrt{2}}|n_1+n_2|,\frac{1}{\sqrt{2}}|n_2-n_2|)^3/0.75$.
3. $\psi=\max(|n_1|,|n_2|)^3/0.75$.

1. For the algorithm to initialize properly, discrete grid directions have to be present.

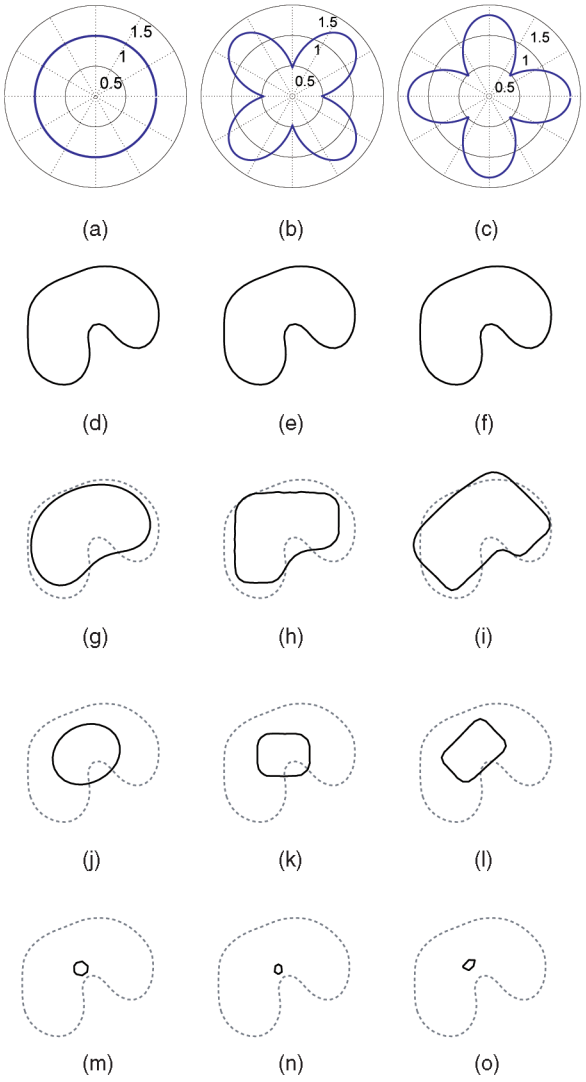


Fig. 1. Synthetic 2D example. These three different local costs depend only on direction. They are represented as polar plots (first row). The corresponding deforming shapes are presented on the following rows (black). The initial curve (gray dashed) is bean shaped. See text.

The first cost is isotropic. In that case, the global cost of the curve is its euclidean length, and the minimizing flow is the euclidean curvature flow [30], [31]. This flow shrinks any planar shape to a circular-shaped point. This is illustrated on the first column in Fig. 1. The second and third costs are defined using direction information. In particular, the second cost favors portions of the curve that are either horizontal or vertical. The third costs does exactly the opposite and favors portions of the curve that are diagonal. The corresponding evolutions can be observed on the second and third columns in Fig. 1. The influence of direction information is very visible in these figures.

5.2 Simulated Example

Finsler active contours extend the isotropic geodesic active contours by adding directionality to the distance functional. Therefore, in oriented domains where directionality is important, Finsler active contours capture important directional information unavailable to isotropic geodesic active contours.

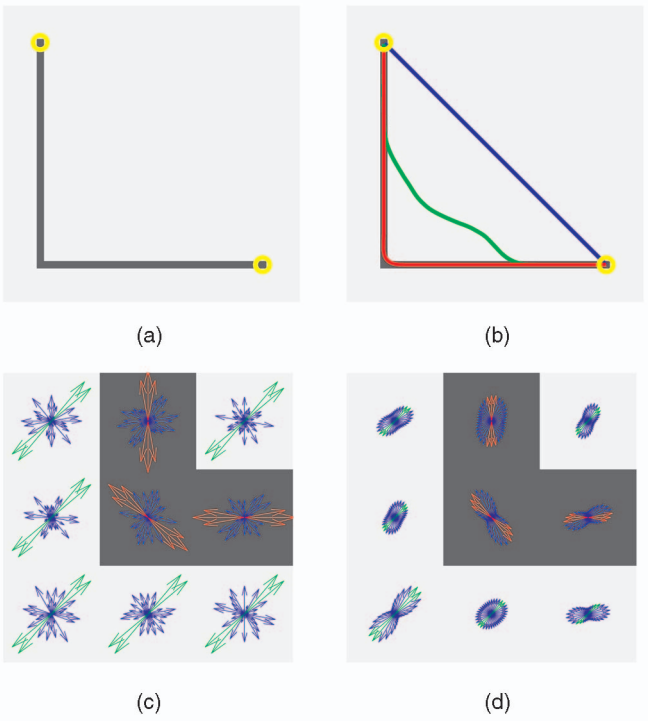


Fig. 2. A simulated 2D example. (a) The synthetic baseline "L" shaped corner. (b) Finsler results (red), Riemannian results (green), and Isotropic results (blue). (c) A zoomed-in view showing the directional data of nine pixels surrounding the corner. (d) A zoomed-in view showing the Riemannian image of nine pixels surrounding the corner, created by applying the Stejskal-Tanner equation to the directional data.

Furthermore, Finsler active contours extend Riemannian active contours by minimizing with respect to the more general Finsler metric. It will be shown that, in some cases, minimization with respect to a Riemannian metric will yield a smoothed version of the result obtained via the Finsler metric.

In order to analyze these properties of Finsler active contours, we have devised a synthetic simulation. We constructed a simple 2D (64×64 pixel) image with an "L" shaped corner, as seen in Fig. 2a. The goal of this simulation is to find an open curve that extends from the top yellow marker to the bottom yellow marker and that follows the "L" shape. We performed this simulation using the dynamic programming numerical scheme. The yellow markers are given as known inputs to the algorithm.

The synthetic directional data was created by randomly drawing samples from the uniform distribution on the interval $[0, 0.5]$ for evenly spaced directions on the unit circle. Then, we added the signal by giving directions of high diffusion slightly stronger values, as shown in Fig. 2c. This figure shows a zoomed-in view of the directional image for nine pixels surrounding the corner. The red arrows correspond to the high-diffusion directions (that is, the signal), the green arrows correspond to a diffusion process contrary to the signal flow, and the blue arrows are the randomly drawn background samples.

Using the Finsler active contour framework, we are able to capture the "L"-shaped corner accurately, as pictured in red in Fig. 2b. We will now compare this to two other methods.

First, we show that adding directional information is critical to capturing the corner. We proceed by comparing the Finsler active contour approach to the isotropic geodesic active contour approach implemented using the Fast Marching Method [17], [18]. At each point in the image, the strongest diffusion value (without respect for directionality) was chosen as the scalar to be used in the isotropic approach. Since, by construction, the strongest diffusion value at each point is the same (that is, the magnitude of red arrows is the same as the magnitude of green arrows), it is obvious that the optimal path for the isotropic geodesic active contour is a straight line connecting the yellow markers, as shown by the blue line in Fig. 2b. This example illustrates the need for directionality, and the difficulties that arise from attempting to discard the directionality via a pointwise scalar function (in this case, the maximum diffusion at each point).

Second, we compared the Finsler active contour approach to the Riemannian active contour approach. In this simulation, we created tensors from the directional data using the Stejskal-Tanner equation, as shown by the zoomed-in view of the corner in Fig. 2d. This figure shows a zoomed-in view of the Riemannian image for nine pixels surrounding the corner. The resulting Riemannian optimal path is shown in green in Fig. 2b. This figure reveals the smoothing effect, which the Riemannian metric tends to have on the result. This is due to the directional averaging, which occurs in the construction of the tensors by imposing the elliptical diffusion profile on the data.

These simulations reveal that, for this particular case, the Finsler active contour is desirable due to the fact that it is capable of capturing a sharp corner in an oriented domain. In other cases, the scalar geodesic active contour or Riemannian active contour approaches may have attributes, which render them more suited to the particular task.

5.3 Curve Detection in Imagery

Given a sample image I for which a portion of a curve Γ^* is given (for example, by a human expert), imagine sliding a small rectangular window along the curve in such a way that the center of the window is always on the curve, and the long axis of the window is aligned with the tangent to the curve. Computing the average value of image intensities at each point inside the window as the window slides along the curve, one obtains an average pattern of what the image looks like locally around the curve Γ^*. A location and direction-dependent pattern detector can then be defined by translating and rotating the average pattern and determining how well it matches the image.

This protocol was applied to a road detection task. Fig. 3 shows the portion of the road that was used to learn the pattern detector. The dimension of the window along its long axis (that is, in the direction of the road) was chosen to be four times the width of the road, and the dimension of the window along its short axis (that is, normal to the road) was chosen to be two times the width of the road. The pattern detector was then obtained for any position and direction by translating and rotating the average window and computing the sum of the square of the difference between the intensity of the image and that of the average window. Curves that will be minimal for this metric will then be those for which the image locally matches the pattern of a road.

Curves were deformed using a straightforward particle-based approach. Fig. 4 shows two different initial curves converging to the same portion of the road. Fig. 4 shows the evolution of a self-intersecting initial curve. Finally, Fig. 5 illustrates the use of dynamic programming. Note that for very low SNR, the dynamic programming fails.

Fig. 3. Road image and manually determined curve used for learning the pattern detector.

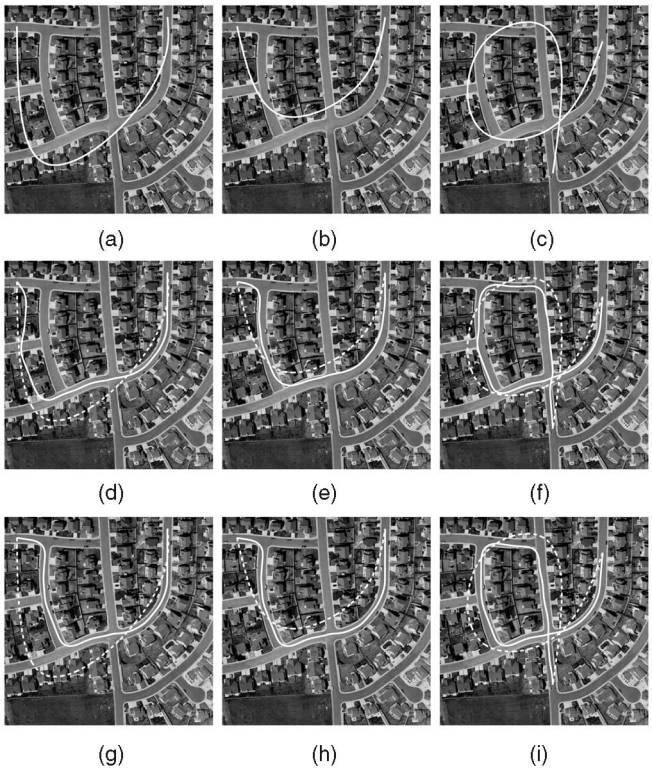


Fig. 4. Particle-based curve evolution with different initializations. (a) Initial curve 1. (b) Initial curve 2. (c) Initial curve 3. (d) Evolving curve 1. (e) Evolving curve 2. (f) Evolving curve 3. (g) Steady state 1. (h) Steady state 2. (i) Steady state 3.

The same experiments were performed on a medical image to track blood vessels. As before, the metric was defined by an initial manual segmentation step. Fig. 6 shows the result of the curve evolution approach. Fig. 7 shows the result of the dynamic programming approach. In that case, noise was artificially added.

5.4 High Angular Diffusion MRI Tractography

Diffusion Tensor Magnetic Resonance Imaging (DT-MRI) measures the diffusion of water in biological tissue [32]. The utility of this method stems from the fact that tissue structure

Fig. 5. Results of road detection on noisy images using dynamic programming. (a) Recovered curve (original). (b) Recovered curve $(\mathrm{SNR} = 0.50)$. (c) Recovered curve $(\mathrm{SNR} = 0.25)$.

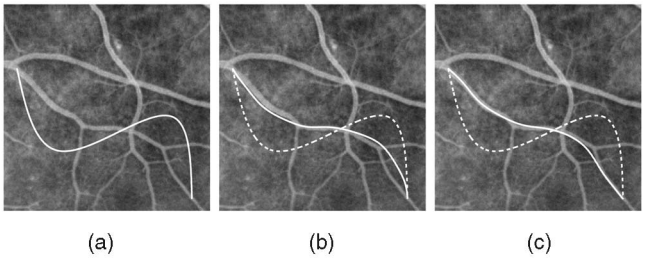


Fig. 6. Curve evolution on a real image. The local cost is determined using a pattern detector. (a) Initial curve. (b) Evolving curve. (c) Steady state.

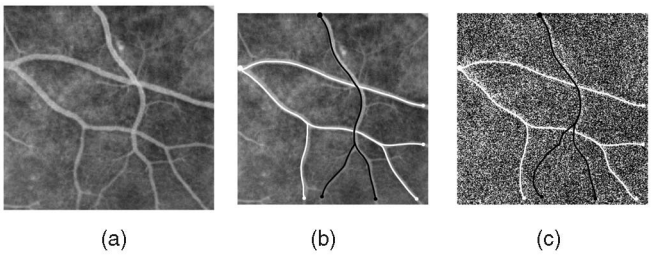
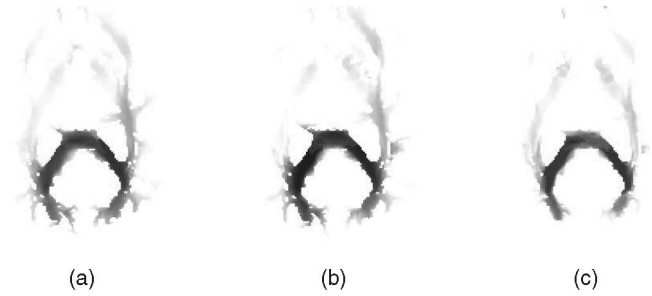


Fig. 7. Vessel detection using dynamic programming. The procedure was run independently for two seed points (large discs) and several target points (small discs). On the right, noise was added. This changes the recovered curves as one of the branches at the bottom is no longer visible. (a) Original image. (b) Recovered curves. (c) Recovered curves $(\mathrm{SNR} = 0.25.)$.

locally affects the Brownian motion of water molecules and will be reflected in the DT-MRI diffusion measurements. In classical theory, diffusion follows a Gaussian process which can be described locally by a second order tensor.

A simple and effective method for tracking nerve fibers using DT-MRI is to follow the direction of maximum diffusion at each voxel [33], [34], [35], [36]. Although this method is wide spread and used in various ways, the fiber trajectory is based solely on local information, which makes it very sensitive to noise. Moreover, the major direction of diffusion can become ill defined, for example, at fiber crossings.

As an application of our framework, tractography is set in a continuous minimum cost framework. This is different from that in [37], [38]. Indeed, in these works, the authors do not propose variational (cost minimizing) techniques. Local costs are defined for every direction on the unit sphere based on high angular resolution diffusion imagery. Equivalently, this can be considered a minimum arrival time framework in which the speed of fictitious particles would be the inverse of the cost.



(a) (b) (c)

Fig. 8. Cost per unit length of end points of optimal curves for different b-values is a validity index. Best results are achieved for the highest b-value. (a) $b = 500$. (b) $b = 1,000$. (c) $b = 1,500$.

5.4.1 Constructing the Direction-Dependent Cost

Most front propagation techniques for diffusion tensor tractography use some *ad hoc* function f of the quadratic form $\hat{\boldsymbol{d}}^t D\hat{\boldsymbol{d}}$, where D is the diffusion tensor. If the Gaussian assumption holds, the diffusion weighted images follow

$$S\left(\boldsymbol{p}, \hat{\boldsymbol{d}}\right) \simeq S(\boldsymbol{p}, \boldsymbol{0}) \exp\left(-b\, \hat{\boldsymbol{d}}^t D(\boldsymbol{p}) \hat{\boldsymbol{d}}\right). \tag{10}$$

Tensor-based techniques can formally be extended to high angular resolution diffusion data sets by setting (see [8] and references therein for details)

$$\psi(\boldsymbol{p}, \hat{\boldsymbol{d}}) \triangleq f\left(-\frac{1}{b}\log\left(\frac{S(\boldsymbol{p}, \hat{\boldsymbol{d}})}{S\boldsymbol{p}, \boldsymbol{0})}\right)\right). \tag{11}$$

However, in the experiments below, we employed the following metric:

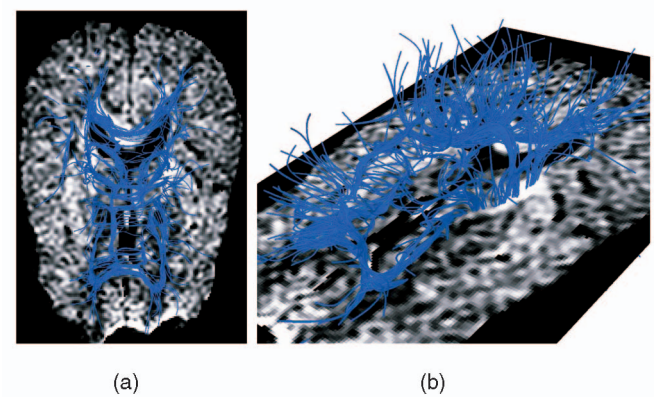
$$\psi(\boldsymbol{p}, \hat{\boldsymbol{d}}) \triangleq \left(\frac{S(\boldsymbol{p}, \hat{\boldsymbol{d}})}{\int_{\hat{\boldsymbol{v}} \perp \hat{\boldsymbol{d}}} \frac{S(\boldsymbol{p}, \hat{\boldsymbol{v}})}{S(\boldsymbol{p}, \boldsymbol{0})} \mathrm{d}\hat{\boldsymbol{v}}}\right)^3. \tag{12}$$

This quantity will be small if there is diffusion in direction $\hat{\boldsymbol{d}}$ (numerator small) and limited diffusion in directions normal to $\hat{\boldsymbol{d}}$ (denominator large). The main advantage of this formulation is that several data points are used to compute the denominator that improves the signal to noise ratio. We chose $f(x) = x^3$ experimentally to accentuate the anisotropy of the data. Because, experimentally, only a few dozen directions are used for acquisition, interpolation was also performed.

It is very important to note that the anisotropic conformal factor ψ is constructed from the data, and for example, in the DW-MRI case, we have no proof that the corresponding F^2 is always strictly convex. However, in Appendix A below, we include for completeness a standard argument that shows that using a scheme such as fast sweeping, one computes the optimum relative to the convexification of F^2. This type of convexification argument is well known in the optimal control literature (see the classical text [39] for details). Thus, we are computing in fact geodesic active contours relative to the Finsler metric defined by the convexification of the defining function.

5.4.2 Results

Here, we show results obtained by applying the methodology described in the above sections to diffusion weighted data sets acquired using a single-shot diffusion-weighted EPI sequence with 31 different gradient directions with b-values (see (10)) of 500, 1,000, and 1,500s/mm^2, on a 1.5 Tesla GE Echospeed system. The data was acquired with different b-values to enable comparisons of the results. Traditional eigenvector based tractography is normally carried out in data with b-values in the range of 700-1,000 s/mm^2. Higher b-values give data with higher angular contrast but at the expense of more noise.



(a) (b)

Fig. 9. Fiber tracking from high angular resolution data set $(\mathrm{b} = 1,500 s/mm^2)$.

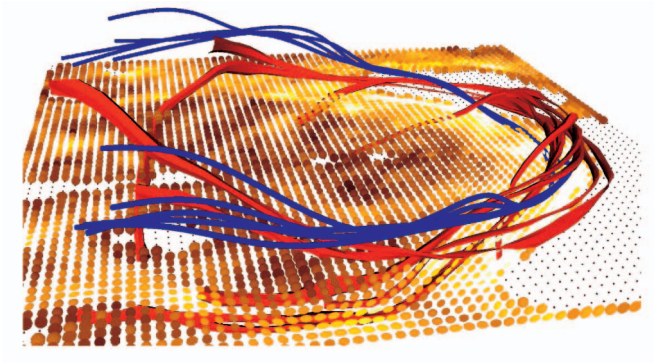


Fig. 10. Proposed technique on high angular resolution data (blue) compared with streamline technique on tensor field (red) $(\mathrm{b} = 1,500 s/mm^2)$.

Cost per unit length, which can be interpreted as a validity index for the putative tracts was determined for all b-values, as shown in Fig. 8.

All curves are optimal given their starting point. The cost per unit length is a measure of the likelihood that a tract from the seed region passes through a given point in the domain. The best contrast (corresponding to the most coherent set of "superoptimal" tracts for a given seed point posterior of the corpus callosum) was obtained at the highest b-value available. This could indicate that the algorithm was able to take advantage of the higher angular contrast in spite of the lower SNR. Tract results for several user-defined seed points are presented in Fig. 9.

Finally, the proposed technique was compared to a streamline technique (see [33], [34], [35], [36]), which requires the computed tensor field, as shown in Fig. 10.

Although validation is a very challenging task due to the lack of ground truth, it can be noted that both algorithms give similar results even though their inputs are different. The tracts of the proposed technique tend to be more coherent as any noise in the data might set the streamline off course, whereas the proposed technique is more global.

5.5 A Note on Timings

Here, we present a note on the timings for each of the experiments. All of the experiments were performed on a common PC. We used a Dell Optiplex GX270 with an Intel Pentium 4 single core chip and 2 GB of memory. Each of the experiments above was conducted using Matlab code with C mex functions for the Fast Sweeping implementation. Moreover, although this code was sufficiently fast for our purposes, we are in the process of porting the code to the freely available Insight Toolkit (ITK) [40].

All of the particle-based approaches, on 2D roads and vessels, converged quickly and in negligible time. Also, all of the Fast Sweeping approaches converged in 3 iterations or less (where one iteration consists of all of the possible directional sweeps through the image), see Section 4.2 for more discussion on Fast Sweeping convergence. For example, it took 0.13 seconds for the experiment in Section 5.2 to converge on a 64×64 grid, and it took 20 seconds for the experiment in Section 5.3 to converge on a 787×787 grid. Also, with simple masking of irrelevant voxels, the time to converge for the 3D DWMRI experiment in Section 5.4 was under 5 minutes. The time to compute a path from a target point back to the seed point is negligible compared to the time required to run the Fast Sweeping portion of the algorithm.

6 Conclusions

In this paper, we proposed a natural approach for adding directionality to the conformal active contour technique. The cost of a curve is defined as the length of the curve weighted by some position and direction-dependent local costs based on image information. This allows for the asymmetric processing of information based on direction. The local costs can be defined from a direction-dependent pattern detector, which can be obtained after a learning step.

The techniques described in the paper are very general and could be used to extract information from many different types of imagery. They have been applied mostly to medical imaging data sets and, in particular, to images of the brain. In fact, it was the problem of extracting white matter tracts that initially motivated this line of research. In the medical area, it could be also be applied to the extraction of blood vessels from various imaging modalities such as magnetic resonance or computed tomography.

Finally, we have only described the Finsler framework in the case of curves. One can derive and study a similar flow for surfaces. This will be the topic of our future research in studying directional-based segmentation methods.

Appendix A

Finsler Computation for Nonconvex Functions

As we noted above, there may be problems in the nonconvex case for our directional segmentation scheme. Nevertheless, as we will indicate in this appendix, the fast-sweeping type numerical approach will automatically capture an approximation for the convexification of the functional (In the sense to be made precise below.) We abstract the situation to be studied as follows.

Let $\psi : S^{n-1} \to \mathbf{R}$ be a function and suppose one defines the *cost of a curve* Γ to be

$$C(\Gamma) = \int_\Gamma \psi(\mathbf{T})\, \mathrm{d}s.$$

Define $F : \mathbf{R}^n \to \mathbf{R}$ to be the homogeneous extension of degree one of ψ, so

$$F(v) = |v|\, \psi\left(\frac{v}{|v|}\right).$$

This function need not be convex. We define its convex hull to be

$$\tilde{F}(v) = \sup\left\{a \cdot v + b : a \in \mathbf{R}^n, b \in \mathbf{R}, \forall_x\ a \cdot x + b \leq F(x)\right\}. \quad (13)$$

Furthermore, we define

$$E = \left\{v \in \mathbf{R}^n : F(v) = \tilde{F}(v)\right\}.$$

This is the set of *extreme points.*

The cost $C(\Gamma)$ of any parameterized curve $\Gamma : [0,1] \to \mathbf{R}^n$ is given by

$$C(\Gamma) = \int_0^1 F(\Gamma'(\xi))\, \mathrm{d}\xi. \quad (14)$$

One can also define the *relaxed cost* as

$$\tilde{C}(\Gamma) = \int_0^1 \tilde{F}(\Gamma'(\xi))\, d\xi. \quad (15)$$

Clearly, one always has

$$\tilde{C}(\Gamma) \leq C(\Gamma) \quad (16)$$

since $\tilde{F}(v) \leq F(v)$ for all $v \in \mathbf{R}^n$.

For any given set $\Sigma \subset \mathbf{R}^n$ and point $p \in \mathbf{R}^n \setminus \Sigma$ one defines the cost to get to Σ from p as

$$C_*(p) = \inf\left\{C(\Gamma) : \Gamma(0) = p, \Gamma(1) \in \Sigma\right\}. \quad (17)$$

Here, the infimum is taken over all curves from p to some point in Σ.

One can also define

$$\tilde{C}_*(p) = \inf\left\{\tilde{C}(\Gamma) : \Gamma(0) = p, \Gamma(1) \in \Sigma\right\}, \quad (18)$$

where the infimum is again taken over all curves from p to some point in Σ.

Lemma 2. *For any curve $\Gamma : [0,1] \to \mathbf{R}^n$ and any $\varepsilon > 0$, there exists a piecewise linear (PL) curve $\tilde{\Gamma} : [0,1] \to \mathbf{R}^n$ with the same endpoints for which one has*

$$C(\tilde{\Gamma}) = \tilde{C}(\tilde{\Gamma}) \leq \tilde{C}(\Gamma) + \varepsilon.$$

This lemma says that any curve from p to Σ can be replaced by a curve with the same endpoints whose velocities are extreme points for F (that is, $\tilde{F}(\Gamma'(\xi)) = F(\Gamma'(\xi))$ for all ξ), without increasing the cost by more than ε.

An immediate consequence of the lemma is:

Lemma 3. $\tilde{C}_*(p) = C_*(p)$.

It follows that any (correct) method that computes C_* by propagating the front $\partial\Sigma$ outwards with velocities given by F will actually compute $\tilde{C}_*$.

Proof of Lemma 2. First, it is clear that one can approximate the given curve Γ by a PL curve Γ_1 for which $\tilde{F}(\Gamma_1'(\xi)) \leq \tilde{F}(\Gamma'(\xi)) + \varepsilon$ holds for $0 \leq \xi \leq 1$. Thus, $\tilde{C}(\Gamma_1) \leq \tilde{C}(\Gamma) + \varepsilon$.

Next, the PL curve Γ_1 is linear (that is, $\Gamma_1'(\xi)$ is constant) on each interval (ξ_{i-1}, ξ_i) from some partition $0 = \xi_0 < \xi_1 < \cdots < \xi_n = 1$. Let v_i be the constant value of Γ_1' on (ξ_{i-1}, ξ_i). If $v_i \notin E$ (recall that E is the set of extreme points), then v_i is a convex combination of certain $\tilde{v}_i^1, \ldots, \tilde{v}_i^n \in E$, that is

$$v_i = \alpha_1 \tilde{v}_i^1 + \cdots + \alpha_n \tilde{v}_i^n, \quad \alpha_i \geq 0, \quad \sum \alpha_i = 1, \tag{19}$$

whereas

$$\tilde{F}(v_i) = \alpha_1 \tilde{F}(v_i^1) + \cdots + \alpha_n \tilde{F}(v_i^n). \tag{20}$$

Now, define a PL curve Γ_2 that has

$$\Gamma_2'(\xi) = v_i^k \text{ for } \xi \in \left(\xi_i^{j-1}, \xi_i^j\right), \tag{21}$$

where $\xi_i^j = \xi_{i-1} + \alpha_j(\xi_i - \xi_{i-1})$. Thus, we replace the segments of Γ_1 whose velocity are not in the extreme set E of the function by PL zigzag curves with the same begin and end points whose velocities are in E.

With this definition, one has

$$\Gamma_2(\xi_i) - \Gamma_2(\xi_{i-1}) = \Gamma_1(\xi_i) - \Gamma_1(\xi_{i-1}).$$

Hence, if one sets $\Gamma_2(0) = \Gamma_1(0) = p$, then one ends up with $\Gamma_2(1) = \Gamma_1(1) \in \Sigma$.

Using (20), one can easily see that $\tilde{C}(\Gamma_2) = \tilde{C}(\Gamma_1)$. Since $\Gamma_2'(\xi) \in E$ for all ξ one also has $C(\Gamma_2) = \tilde{C}(\Gamma_2)$. Hence, Γ_2 is a curve from p to Σ with $C(\Gamma_2) \leq \tilde{C}(\Gamma) + \varepsilon$.□

APPENDIX B

FINSLER FLOW IN TERMS OF F

In this section, we describe the Finsler flow in terms of the homogeneous extension F and derive some of its properties.

B.1 First Variation Using F

Instead of writing $L(\Gamma)$ in terms of ψ, we can also write $L(\Gamma)$ as in (4). The first variation is then given by the usual Euler-Lagrange equation

$$\frac{d}{dt} L(\Gamma) = \int_0^1 \left\{F_p - (F_v)_x\right\} \cdot \Gamma_t \, \mathrm{d}x. \tag{22}$$

If one looks for a normal evolution equation (that is, $\Gamma_t \perp \Gamma_x$), then one is led to an equation of the form

$$\beta \Gamma_t = \left\{(F_v)_x - F_p\right\}^\perp$$

for some positive scalar β. If one additionally wants the equation to be invariant under reparametrization, then the only possible choice for β is $\beta = G(\Gamma, \Gamma_x)$ in which $G(p, v)$ is positively homogeneous of degree one in $v \in \mathbf{R}^n$. A possible choice would be $G(\Gamma, \Gamma_x) = |\Gamma_x|$, which leads us to the evolution equation

$$\Gamma_t = \frac{1}{|\Gamma_x|} \left\{(F_v)_x - F_p\right\}^\perp. \tag{23}$$

This equation is equivalent with (5).

One could also choose $G(p, v) = F(p, v)$, which would result in

$$\Gamma_t = \frac{1}{F(\Gamma, \Gamma_x)} \left\{(F_v)_x - F_p\right\}^\perp.$$

B.2 Some Identities Involving F

Since $F(p, tv) = tF(p, v)$, for all $t \geq 0$, one has

$$F_v(p, tv) = F_v(p, v) \qquad (\forall t > 0), \tag{24}$$

$$F_v(p, v) \cdot v = 0. \tag{25}$$

For the second derivative $F_{vv}(p, v)$, which we regard as a symmetric linear transformation on $\mathbf{R}^n$, this implies that

$$F_{vv}(p, v) \cdot v = 0 \tag{26}$$

and, hence,

$$F_{vv}(p, v) \cdot w \perp v \qquad (\forall w \in \mathbf{R}^n). \tag{27}$$

We may also regard $F_{vp}(p, v)$ as a linear transformation on $\mathbf{R}^n$, and in this case, we have

$$\left\{v \cdot F_{pv}(p, v) - F_p(p, v)\right\} \perp v. \tag{28}$$

Indeed, in tensor notation, this amounts to

$$v_i F_{p_i v_j}(p, v) v_j = F_{p_i}(p, v) v_i,$$

which one obtains by differentiating the Euler identity

$$F_{v_i}(p, v) v_i = F(p, v)$$

with respect to p in the direction of v.

B.3 Steepest Descent with F

We continue with (23)

$$|\Gamma_x| \, \Gamma_t = \left\{F_{vv}(\Gamma, \Gamma_x)\Gamma_{xx} + \Gamma_x \cdot F_{pv}(\Gamma, \Gamma_x) - F_p(\Gamma, \Gamma_x)\right\}^\perp.$$

By (27), the combined terms on the right are already perpendicular to Γ_x. We therefore find that (23) is equivalent with

$$\Gamma_t = \frac{1}{|\Gamma_x|} \left\{F_{vv}(\Gamma, \Gamma_x) \cdot \Gamma_{xx} + \Gamma_x \cdot F_{pv}(\Gamma, \Gamma_x) - F_p(\Gamma, \Gamma_x)\right\}. \tag{29}$$

More generally, one gets the equation

$$\Gamma_t = \frac{1}{G(\Gamma, \Gamma_x)} \left\{F_{vv}(\Gamma, \Gamma_x) \cdot \Gamma_{xx} + \Gamma_x \cdot F_{pv}(\Gamma, \Gamma_x) - F_p(\Gamma, \Gamma_x)\right\}. \tag{30}$$

No matter which G one chooses, this equation fails to be parabolic since F_{vv} always has a zero eigenvalue, namely, $F_{vv}(p, v)v = 0$.

B.4 Parabolic Equation

The right-hand side in (30) is invariant under reparametrizations, that is, if $\Gamma(x, t) = \gamma(y(x, t), t)$, then γ satisfies

$$\gamma_t = \frac{1}{G(\gamma, \gamma_y)} F_{vv}(\gamma, \gamma_y)\gamma_{yy} + y_t \gamma_y.$$

Choose the parametrization so that

$$y_t(x, t) = \left[\alpha[\gamma](\gamma_y \cdot \gamma_{yy})\gamma_y\right]_{y=y(x,t)}$$

for some scalar $\alpha > 0$, which can depend on Γ and Γ_y.

The resulting equation for γ is then

$$\gamma_t = \left\{ \frac{F_{vv}(\gamma, \gamma_y)}{G(\gamma, \gamma_y)} + \alpha \gamma_y \otimes \gamma_y \right\} \cdot \gamma_{yy} + \frac{B(\gamma, \gamma_y)}{G(\gamma, \gamma_y)}, \tag{31}$$

where by definition

$$B(p, v) = v \cdot F_{pv}(p, v) - F_p(p, v).$$

As long as one chooses $\alpha(\gamma, \gamma_y) > 0$, and as long as F_{vv} is positive definite on $\{v\}^{\perp}$, this equation is strictly parabolic. A particular choice for G and α would be

$$G = |\gamma_y|, \quad \alpha = |\gamma_y|^{-2},$$

which leads to

$$\Gamma_t = \left\{ F_{vv}(\gamma, \mathbf{T}) + \mathbf{T} \otimes \mathbf{T} \right\} \cdot \frac{\gamma_{yy}}{|\gamma_y|^2} + B(\gamma, \mathbf{T}), \tag{32}$$

where $\mathbf{T} = \gamma_y / |\gamma_y|$ is the unit tangent vector.

B.5 Numerical Scheme

We should note that using the above, a simple approach can be employed to (32). For completeness, we sketch this here.

We set

$$\gamma(j\Delta x, k\Delta t) = \gamma_j^k$$

and discretize (32) as follows:

$$\frac{\gamma_j^{k+1} - \gamma_j^k}{\Delta t} = A_j^k \left\{ \gamma_{j+1}^{k+1} - 2\gamma_j^{k+1} + \gamma_{j-1}^{k+1} \right\} + B_j^k \tag{33}$$

in which the $n \times n$ matrices A_j^k are defined by

$$A_j^k = 4 \frac{F_{vv}(\gamma_j^k, \mathbf{T}_j^k) + \mathbf{T}_j^k \otimes \mathbf{T}_j^k}{|\gamma_{j+1}^k - \gamma_{j-1}^k|^2}$$

and one could define the unit tangents $\mathbf{T}_j^k$ by

$$\mathbf{T}_j^k = \frac{\gamma_{j+1}^k - \gamma_{j-1}^k}{|\gamma_{j+1}^k - \gamma_{j-1}^k|}.$$

The vectors B_j^k could be discretized by

$$B_j^k = \mathbf{T}_j^k \cdot F_{pv}(\gamma_j^k, \mathbf{T}_j^k) - F_p(\gamma_j^k, \mathbf{T}_j^k).$$

The system of equations (33) is a tridiagonal vector-valued system of equations. If one puts the components of each vector γ_j^k in one long vector, then (33) can be written as a finite band system of equations, which can be solved very efficiently ($O(n)$ operations per time step).

ACKNOWLEDGMENTS

Allen Tannenbaum was supported by a Marie Curie Grant through the EU. This work was supported in part by grants from NSF, AFOSR, ARO, MURI, MRI-HEL as well as by a grant from NIH (NAC P41 RR-13218) through Brigham and Women's Hospital. This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by the US National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. Information on the National Centers for Biomedical Computing can be obtained from http://nihroadmap.nih.gov/bioinformatics.

REFERENCES

[1] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Conformal Curvature Flows: From Phase Transitions to Active Vision," *Archive of Rational Mechanics and Analysis,* vol. 134, no. 3, pp. 275-301, 1996.

[2] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," *Int'l J. Computer Vision,* vol. 22, no. 11, pp. 61-79, 1997.

[3] D. Bao, S.-S. Chern, and Z. Shen, *Introduction to Riemann-Finsler Geometry.* Springer, 2000.

[4] M. Spivak, *Introduction to Differential Geometry,* vol. 2. Publish of Perish, 1974.

[5] S. Angenent and M. Gurtin, "Anisotropic Motion of a Phase Interface. Well-Posedness of the Initial Value Problem and Qualitative Properties of the Interface," *J. Reine und Angewandte Mathematik,* vol. 446, pp. 1-47, 1994.

[6] M. Gage, "Evolving Plane Curves by Curvature in Relative Geometries," *Duke J. Math.,* vol. 72, pp. 441-466, 1993.

[7] E. Pichon, G. Sapiro, and A. Tannenbaum, Segmentation of Diffusion Tensor Imagery, "Directions in Math. Systems Theory and Optimization," *Lecture Notes in Computer Science,* no. 286, pp. 239-247, 2003.

[8] E. Pichon, C. Westin, and A. Tannenbaum, "Hamilton-Jacobi-Bellman Approach to High-Angular Resolution Diffusion Tractography," *Proc. Int'l Conf. Medical Image Computing and Computer-Assisted Intervention,* pp. 180-187, 2005.

[9] E. Pichon and A. Tannenbaum, "Pattern Detection and Image Segmentation with Anisotropic Conformal Factor," *Proc. Int'l Conf. Image Processing,* 2005.

[10] G. Bellettini, "Anisotropic and Crystalline Mean Curvature Flow," *A Sampler of Riemann-Finsler Geometry,* pp. 49-82, Cambridge Univ. Press, 2004.

[11] R. Kimmel and A. Bruckstein, "Regularized Zero-Crossings as Optimal Edge Detectors," *Int'l J. Computer Vision,* vol. 53, no. 3, pp. 225-243, 2003.

[12] A. Vasilevsky and K. Siddiqi, "Flux Maximizing Geometric Flows," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 12, pp. 1565-1579, Dec. 2002.

[13] Y. Boykov, V. Kolmorgorov, D. Cremers, and A. Delong, "An Integral Solution to Surface Evolution Pdes via Geo-Cuts," *Proc. IEEE European Conf. Computer Vision,* pp. 409-422, 2006.

[14] V. Kolmorgorov and Y. Boykov, "What Metrics Can Be Approximated by Geo-Cuts or Global Optimization of Length/Area and Flux," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[15] Y. Boykov and V. Kolmorgorov, "Computing Geodesics and Minimal Surfaces via Graph Cuts," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 26-33, 2003.

[16] J.-M. Morel and S. Solimini, *Variational Methods for Image Segmentation.* Birkhauser, 1994.

[17] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces.* Springer, 2003.

[18] J. Sethian, *Level Set Methods and Fast Marching Methods.* Cambridge Univ. Press, 1999.

[19] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis.* Cambridge Univ. Press, 2001.

[20] E. Mortensen, B. Morse, W. Barrett, and J. Udupa, "Adaptive Boundary Detection Using Live-Wire Two-Dimensional Dynamic Programming," *IEEE Proc. Computers in Cardiology,* pp. 635-638, 1992.

[21] J.N. Tsitsiklis, "Efficient Algorithms for Globally Optimal Trajectories," *IEEE Trans. Automatic Control,* vol. 50, no. 9, pp. 1528-1538, 1995.

[22] S. Angenent, "Parabolic Equations for Curves on Surfaces. II. Intersections, Blow-Up and Generalized Solutions," *Ann. Math. (2),* vol. 133, no. 1, pp. 171-215, 1991.

[23] J.A. Oaks, "Singularities and Self-Intersections of Curves Evolving on Surfaces," *Indiana Univ. Math. J.,* vol. 43, no. 3, pp. 959-981, 1994.

[24] M. DoCarmo, *Riemannian Geometry.* Springer, 2003.

[25] W. Fleming and H. Soner, *Controlled Markov Processes and Viscosity Solutions.* Springer, 2003.

[26] H. Soner, "Dynamic Programming and Viscosity Solutions," *Proc. Ann. AMS Meeting,* 1999.

[27] J. Sethian and A. Vladimirsky, "Ordered Upwind Methods for Static Hamilton-Jacobi Equations: Theory and Applications," *SIAM J. Numerical Analysis,* vol. 41, no. 1, pp. 325-363, 2003.

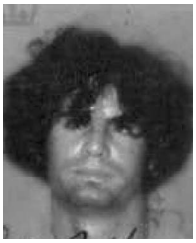
[28] C. Kao, S. Osher, and Y. Tsai, "Fast Sweeping Methods for Static Hamilton-Jacobi Equations," Technical Report 03-75, CAM, Univ. of California, Los Angeles, 2003.

[29] C. Kao, S. Osher, and J. Qian, "Lax-Friedrichs Sweeping Scheme for Static Hamilton-Jacobi Equations," *J. Computational Physics,* vol. 196, no. 1, pp. 367-391, May 2004.
[30] M. Gage and R. Hamilton, "The Heat Equation Shrinking Convex Plane Curves," *J. Differential Geometry,* vol. 23, pp. 69-96, 1986.
[31] M. Grayson, "The Heat Equation Shrinks Embedded Plane Curves to Round Points," *J. Differential Geometry,* vol. 26, pp. 285-314, 1987.
[32] P. Basser, J. Mattiello, and D. LeBihan, "MR Diffusion Tensor Spectroscopy and Imaging," *Biophysical J.,* vol. 66, pp. 259-267, 1994.
[33] S. Mori, B. Crain, V. Chacko, and P. van Zijl, "Three-Dimensional Tracking of Axonal Projections in the Brain by Magnetic Resonance Imaging," *Annals of Neurology,* vol. 45, no. 2, pp. 265-269, Feb. 1999.
[34] T. Conturo, N. Lori, T. Cull, E. Akbudak, A. Snyder, J. Shimony, R. McKinstry, H. Burton, and M. Raichle, "Tracking Neuronal Fiber Pathways in the Living Human Brain," *Proc. Nat'l Academy of Sciences,* pp. 10422-10427, Aug. 1999.
[35] C.-F. Westin, S.E. Maier, B. Khidhir, P. Everett, F.A. Jolesz, and R. Kikinis, "Image Processing for Diffusion Tensor Magnetic Resonance Imaging," *Proc. Int'l Conf. Medical Image Computing and Computer-Assisted Intervention,* pp. 441-452, 1999.
[36] P. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi, "In Vivo Fiber Tractography Using DT-MRI Data," *Magnetic Resonance in Medicine,* vol. 44, pp. 625-632, 2000.
[37] P. Hagmann, T.G. Reese, W.-Y.I. Tseng, R. Meuli, J.-P. Thiran, and V.J. Wedeen, "Diffusion Spectrum Imaging Tractography in Complex Cerebral White Matter: An Investigation of the Centrum Semiovale," *Proc. Int'l Soc. Magnetic Resonance in Medicine,* 2004.
[38] J.S. Campbell, "Diffusion Imaging of White Matter Fibre Tracts," PhD dissertation, McGill Univ., 2004.
[39] L.C. Young, *Calculus of Variations and Optimal Control Theory.* W.B. Saunders, 1969.
[40] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, "The ITK Software Guide. Kitware," technical report, http://www.itk.org/ItkSoftwareGuide.pdf, 2003.

John Melonakos received BS degree in electrical engineering from Brigham Young University in 2004. He is currently working toward the PhD degree in electrical and computer engineering at the Georgia Institute of Technology. He is currently researching on mathematical methods for computer vision and image processing with Professor Allen Tannenbaum. He has won several fellowships, including the US National Science Foundation sponsored Technological Innovation: Generating Economic Results Fellowship, the NSF sponsored Hybrid Neural Microsystems Fellowship, the Sam Nunn Security Program Fellowship, and the Georgia Tech President's Fellowship. He is a student member of the IEEE.

Eric Pichon received the Diplome d'Ingenieur degree from Ecole Superieure d'Electricite (Supelec) in 2001 and the PhD degree in electrical and computer engineering from the Georgia Institute of Technology in 2005. His research interest includes partial differential techniques for computer vision with an emphasis on medical imaging. He is currently with GE Healthcare. He is a member of the IEEE.

Sigurd Angenent received the PhD degree in mathematics from the University of Leiden in 1986. He has been in the faculty of the department of mathematics at the University of Wisconsin since 1988 and works on nonlinear heat equations in differential geometry, biology, and image processing. He is a member of the Royal Netherlands Academy of Sciences. He is a member of the IEEE.



Allen Tannenbaum is a faculty member of the School of Electrical and Computer Engineering and the School of Biomedical Engineering of the Georgia Institute of Technology. His research interests include image processing, control, and computer vision. He is a member of the IEEE.